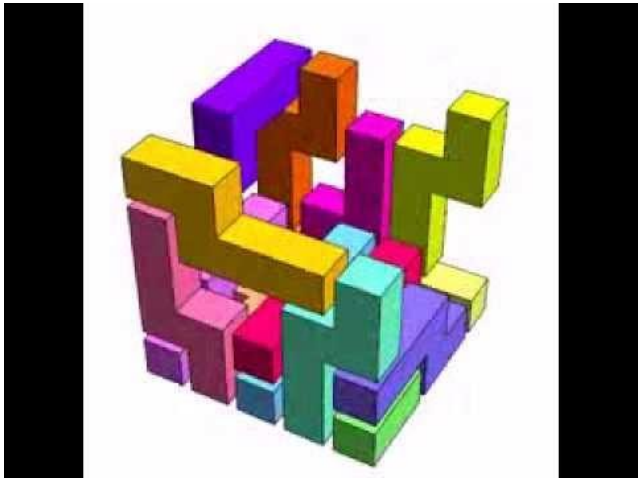

PHASE 3
GROUP 20

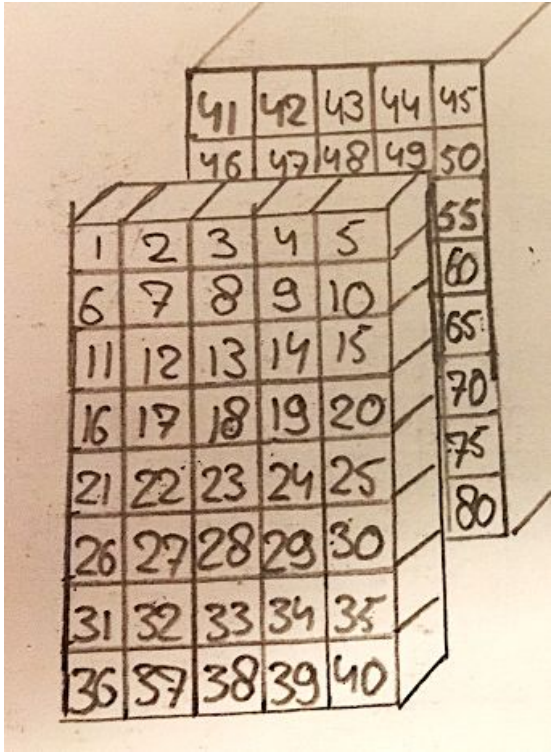
Introduction

- Knapsack 0-1 problem
- Research for algorithms
- Set a plan
- Task assignment
- Visualization (JavaFX)
- Meet the deadlines

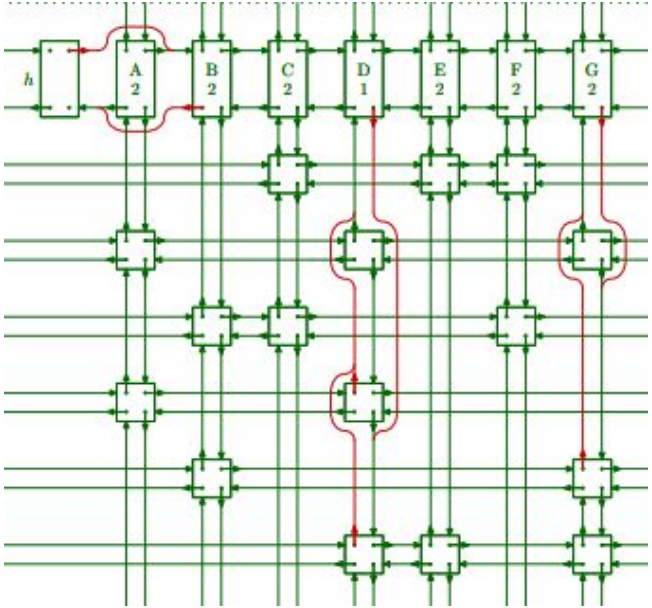


Matrix creation

- 3 arbitrary dimensions
- Flipping the dimensions, not the piece
- Check if it is in the container
- Computation from x, y and z to index



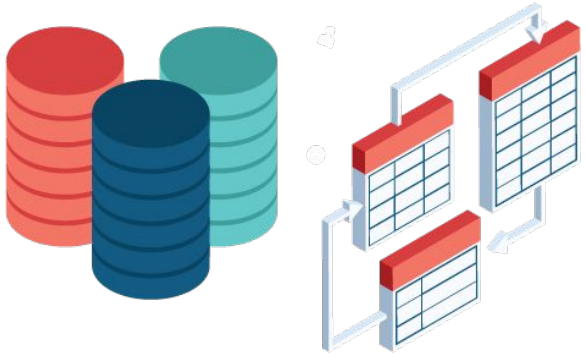
Basic Dancing Links Algorithm



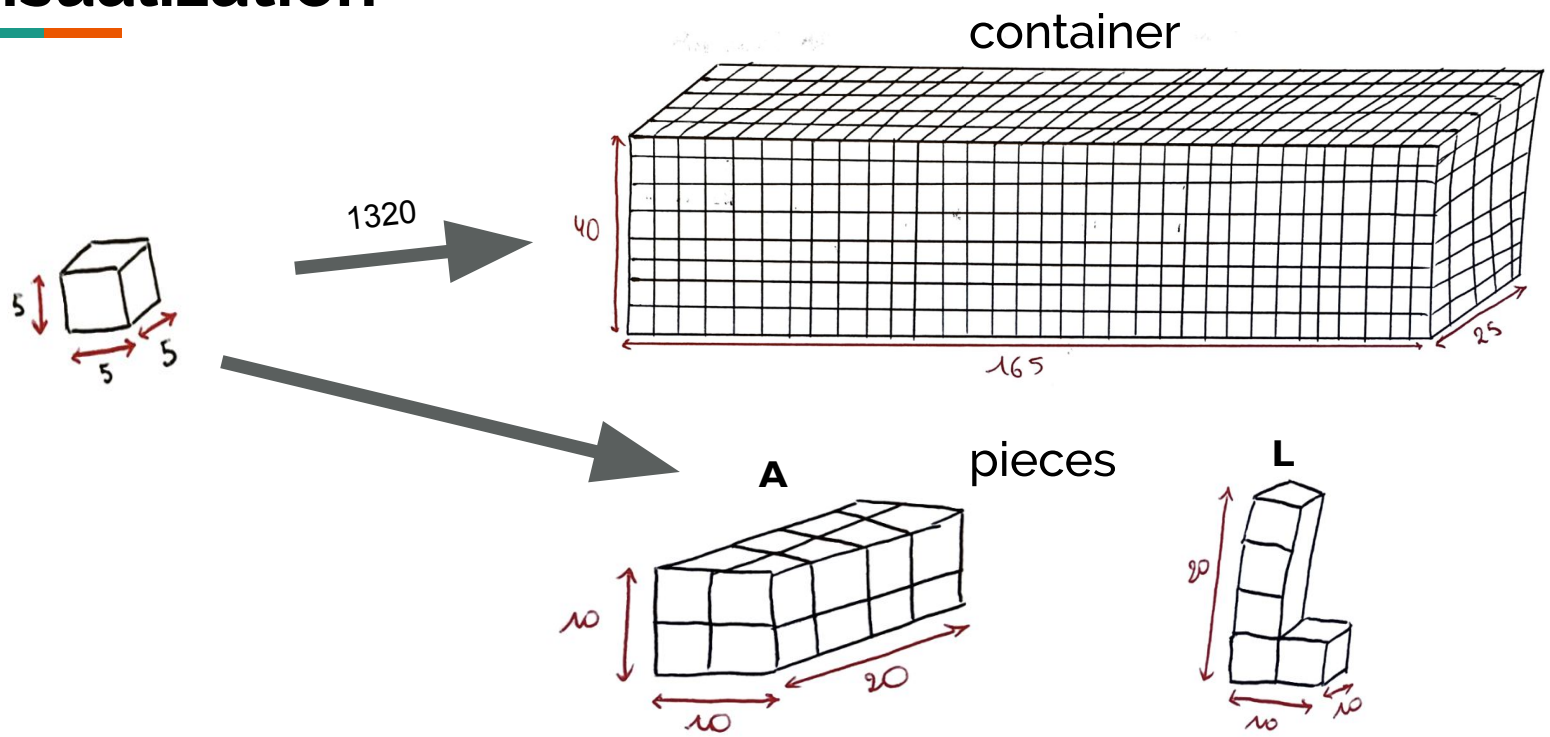
- Column Contains Valid Piece,
  - New Node Instance Field PieceID
- Pick Valid Piece
  - Trackers
- If Current Solution > Stored Solution
  ->Store New Solution
- Search Time Limit

Algorithm Modifications/Optimizations

1. Basic Algorithm
2. DataBase Restructuring
3. Divide and Conquer



Visualization



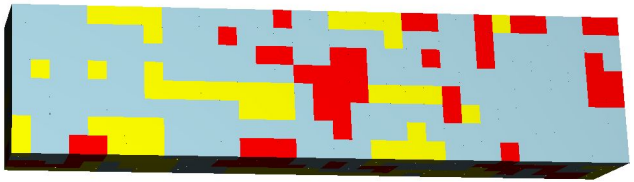
- Draw the truck and pieces
- Play with rotation & colors

\+ Mouse listener 

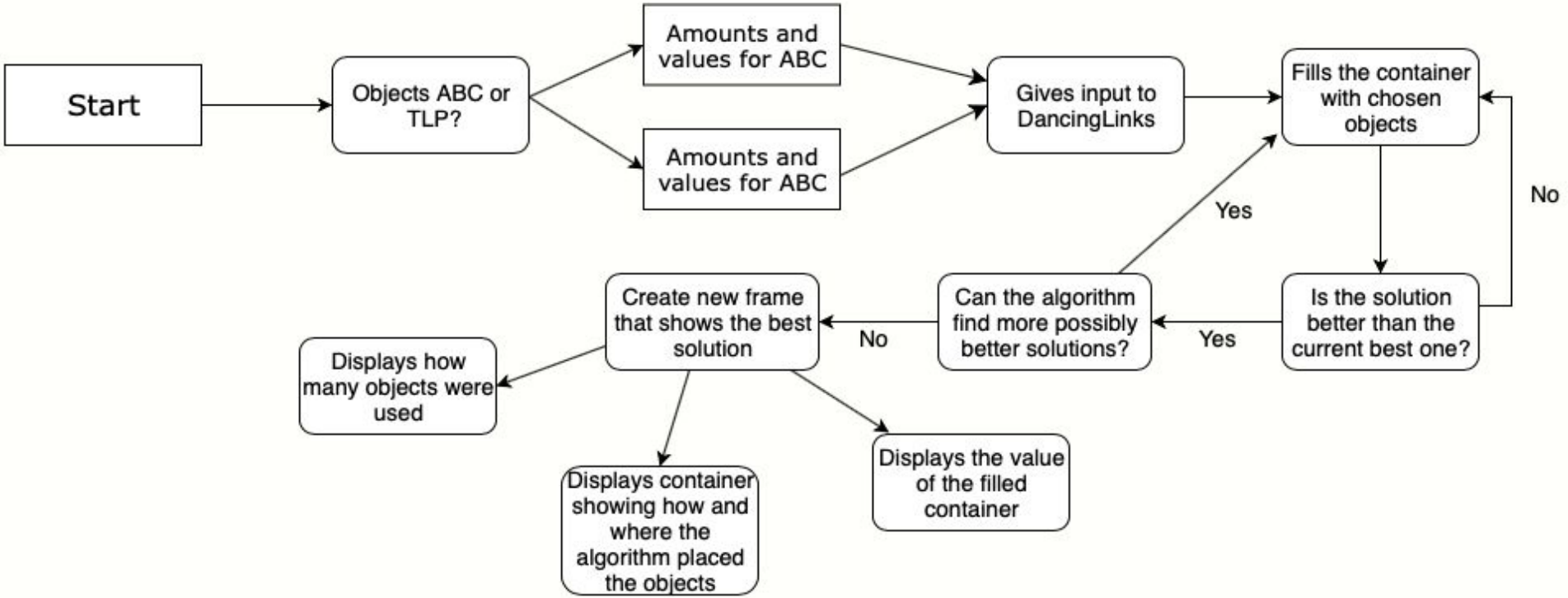
\+ Key listener 

FIELD

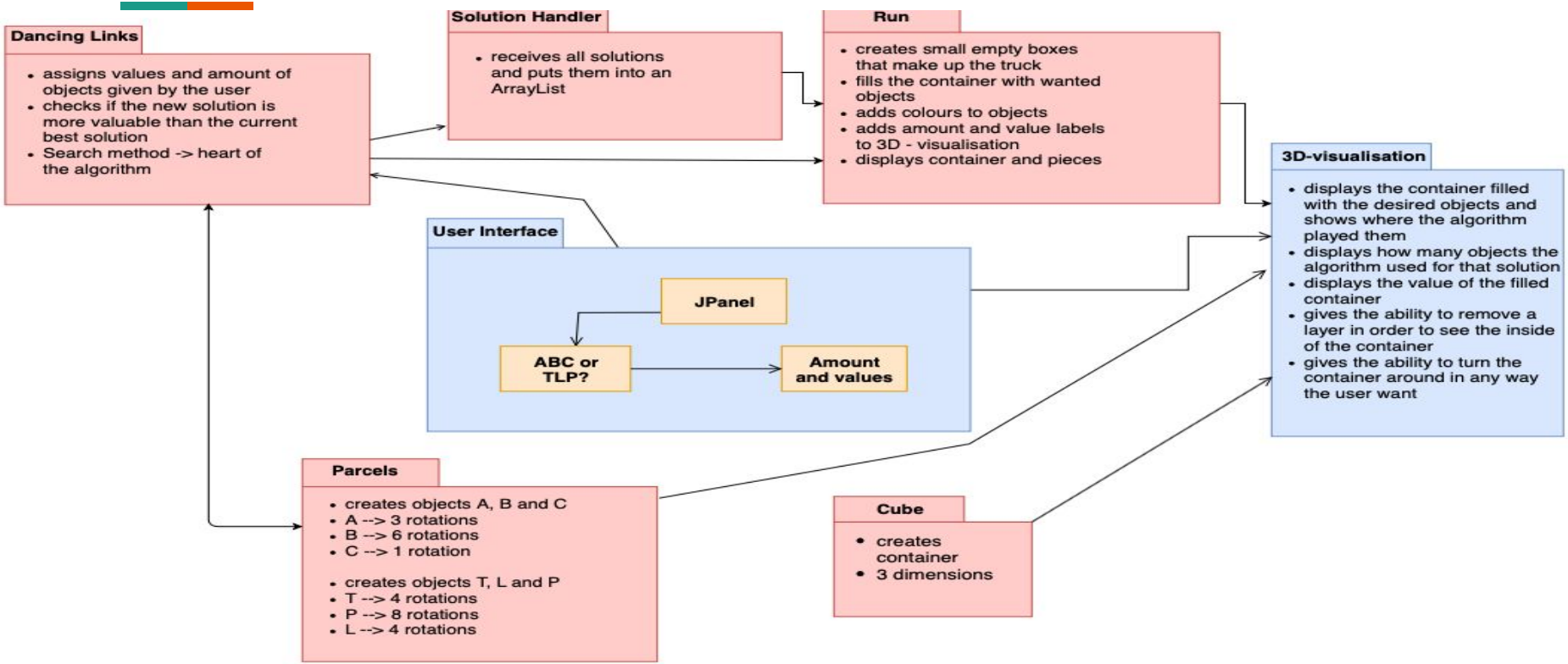
0: Empty
1: Blue
2: Red
3: Yellow



Flowchart


Start
Objects ABC or TLP?
Amounts and values for ABC
Amounts and values for ABC
Gives input to DancingLinks
Fills the container with chosen objects
No
Yes
Is the solution better than the current best one?
Yes
Can the algorithm find more possibly better solutions?
No
Create new frame that shows the best solution
Displays how many objects were used
Displays container showing how and where the algorithm placed the objects
Displays the value of the filled container

UML Diagram


Dancing Links
assigns values and amount of objects given by the user
checks if the new solution is more valuable than the current best solution
Search method -> heart of the algorithm
Solution Handler
receives all solutions and puts them into an ArrayList
Run
creates small empty boxes that make up the truck
fills the container with wanted objects
adds colours to objects
adds amount and value labels to 3D - visualisation
displays container and pieces
3D-visualisation
displays the container filled with the desired objects and shows where the algorithm played them
displays how many objects the algorithm used for that solution
displays the value of the filled container
gives the ability to remove a layer in order to see the inside of the container
gives the ability to turn the container around in any way the user want
User Interface
JPanel
ABC or TLP?
Amount and values
Parcels
creates objects A, B and C
A --> 3 rotations
B --> 6 rotations
C --> 1 rotation
creates objects T, L and P
T --> 4 rotations
P --> 8 rotations
L --> 4 rotations
Cube
creates container
3 dimensions

Results

The best solution by Dancing Links algorithm

Parcel type	A	B	C
Quantity	30	10	21
Value	3	4	5

Maximum value: 235

Parcel type	L	P	T
Quantity	16	67	181
Value	3	4	5

Maximum value: 1221

The solution by Dancing Links algorithm (divide-and-conquer)

Parcel type	L	P	T
Quantity	9	78	177
Value	3	4	5

Maximum value: 1224

Conclusion

Questions	Answers
Is it possible to fill the complete cargo space with A, B and/or C parcels, without having any gaps?	✗
If parcels of type A, B and C represent values of 3, 4 and 5 units respectively, then what is the maximum value that you can store in your cargo-space?	235 units
Is it possible to fill the complete cargo space with L, P and/or T parcels, without having any gaps?	✓
If parcels of type L, P and T represent values of 3, 4 and 5 units respectively, then what is the maximum value that you can store in your cargo-space?	1224 units